REMARKS

I.   Introduction

In response to the Office Action dated June 25, 2004, claims 1-8, 10, 12-19, 21, 23, and 24 have amended, claims 9 and 20 have been canceled, and claims 26-28 have been added. Claims 1-8, 10-19, and 21-28 remain in the application. Re-examination and re-consideration of the application is requested.

II.   Non Art Rejection

On page (2) of the Office Action, claims 12-22 were rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Further, claims 12 was rejected under 35 U.S.C. §101 for claiming a recitation of a use, without setting forth any steps involve din the process.

Applicants have amended claim 12 to more clearly claim the subject matter and set fort appropriate steps. Accordingly, Applicants submit that the rejections are now moot.

III.   Prior Art Rejections

A. The Office Action Rejections

On page (3) of the Office Action, claims 1-25 were rejected under 35 U.S.C. §102(b) as being anticipated by Linenbach et al., U.S. Patent No. 5,903,794 (Linenbach).

Applicants' attorney respectfully traverses these rejections.

B. The Applicants' Independent Claims

Independent claims 1, 12, 23, and 24 are generally directed to updating data in a database. Specifically, there are persistent copies of objects in a database. A database application makes a transient copy of the persistent objects and performs modifications to the transient copies. In response to the modifications, a database thread generates database transaction requests that are used to update the persistent copy of the database. The requests are processed in a queue of database transaction requests at a lower priority than the modifications themselves. Accordingly, the user may modify a transient copy of an object and continue accessing the transient copy while the persistent copy of the data is being updated at a lower priority through the database queue.

-6-

G&C 30566.194-US-01

In addition, the new dependent claims provide further limitations with regards to how the transient copies are maintained. Specifically, dependent claims 26-28 provide that when the transient copy of the persistent object in one client is accessed, any previously existing transient copy of the persistent object in another client is unloaded from transient object cache of the other client. Thus, when one transient copy is being accessed, transient copies at other clients are unloaded from cache memory.

C. The Linenbach Reference

A deferred transaction mechanism facilitates multi-threaded operation of database application programs. The deferred transaction mechanism allows data items to be committed from the local memory of a computer system to a database system in a background thread, while other foreground threads continue to read the data item. In most instances, this makes the delay in committing a data item to the database imperceptible to a human user of a database application program. The deferred transaction mechanism further supports an "undo" feature, which allows modifications to a data item located in a computer's local memory to be rapidly discarded.

D. The Applicants' Invention is Patentable Over the References

The Applicants' invention, as recited in independent claims 1, 12, 23, and 24 is patentable over the references, because it contains limitations not taught by the references.

The Office Action, however, asserts the following with respect to the independent claims:

Claims 1, 12 and 23-25:
Linenbach discloses:
- visual display means, processing means, storage means and memory means [Fig 1]
- wherein said memory means is configured to sotre program instructions for updating data in a database, having persistent copies of objects that control processing steps [Fig 1, 160],
- wherein a database application makes modifications to transient copies of said persistent objects [Fig 4, 440];
- a database thread generates database transaction requests in response to said modifications [col 7, lines 10-35];
- said requests are processed at a lower priority than said modifications [Fig 4 and user generated commit signal, col 7 lines 10-35].
-

In addition, original dependent claims 9 and 20 provided for the use of a queue. The Office Action rejected claims 9 and 20 as follows:

-7-

<u>Claims 9 and 20:</u>

    Linenbach discloses wherein said object cache manager queues transactions corresponding to amendments of said transient copies in a database request queue as transaction requests [Fig 3, 314, col 6, lines 15-25]

Applicants' attorney respectfully disagrees. The independent claims have been amended to provide that the database requests for updating the persistent copies of the database are processed in a queue of database transaction requests. In rejecting the original dependent claims that provided for a queue, the Office Action relied on col. 6, lines 15-25 which provides:

> A database application can activate a context 315, deactivate a context 315, or create a new context 315. However, the details of context management, including the context stack 314, are hidden from the database application. The context stack 314 can be thought of as a stack of contexts. As a thread picks up a new context 315, that context 315 is placed at the top of the context stack 314 and becomes the active context. FIG. 3 illustrates a single context 315 connected to context stack 314, although it is possible for a larger number of contexts to be associated with a single context stack 314.

As can be seen, this text merely describes the use of contexts and a stack of contexts. First, Linenbach's context is a data structure associated with a unit of work that may be performed by a group of threads. The context keeps track of which objects a group of threads has accessed, and which objects a group of threads has modified. A context also contains pointers to data items (addresses of data items). (See col. 6, lines 5-15). Such a context is not equivalent to a transaction request that is generated in response to a modification to a transient copy of a database. In this regard, the claimed transaction request is not a unit of work that may be performed by a group of threads. Instead, it is a request to modify a persistent copy of data in response to a modification of a transient copy of the persistent data.

Secondly, the claimed context describes the use of a context stack and not a queue. A stack processes data in FILO (First-In-Last-Out) order, while a queue processes data in FIFO (First-In-First-Out) order. Stacks and queues may be implemented in a variety of ways that are distinguishable from each other. Further, stacks and queues serve different purposes and provide distinctly different results because of the order in which things are processed. In this regard, any disclosure of a stack does not and cannot teach or render obvious a specifically claimed queue.

In addition, the remainder of Linenbach also fails to describe the transaction requests, queue of transactions requests, and processing of such requests as claimed in the present invention.

Further, as described above, the new dependent claims provide for unloading a transient copy of data in one client if another client is processing data on their own transient copy of the data. Such a teaching is not even remotely referred to or suggested, implicitly or explicitly, anywhere in Linenbach.

-8-

G&C 30566.194-US-01

Thus, Applicants' attorney submits that independent claims 1, 12, 23, and 24 are allowable over the references. Further, dependent claims 2-8, 10-11, 13-19, 21-22, and 25-28 are submitted to be allowable over the references in the same manner, because they are dependent on independent claims 1, 12, 23, and 24, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-8, 10-11, 13-19, 21-22, and 25-28 recite additional novel elements not shown by the references.
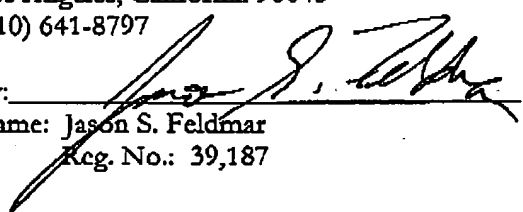
IV.   Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: September 27, 2004

By:
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/

-9-

G&C 30566.194-US-01